

[illegible]

```

HH      HH      000000      MM      MM      000000      GGGGGGGG
HH      HH      000000      MM      MM      000000      GGGGGGGG
HH      HH      00      00      MMMM      MMMM      00      00      GG
HH      HH      00      00      MMMM      MMMM      00      00      GG
HH      HH      00      00      MM      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      MM      00      00      GG
HHHHHHHHHHHH      00      00      MM      MM      00      00      GG
HHHHHHHHHHHH      00      00      MM      MM      00      00      GG
HH      HH      00      00      MM      MM      00      00      GG      GGGGGG
HH      HH      00      00      MM      MM      00      00      GG      GGGGGG
HH      HH      00      00      MM      MM      00      00      GG      GG
HH      HH      00      00      MM      MM      00      00      GG      GG
HH      HH      000000      MM      MM      000000      GGGGGG
HH      HH      000000      MM      MM      000000      GGGGGG

```

```

LL               IIIII
LL               IIIII
LL               II
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LLLLLLLLLLLLLL IIIII
LLLLLLLLLLLLLL IIIII
SSSSSSSSSS
SSSSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSS
SSSSSSSS

```

```
(2)      82      DECLARATIONS
(3)      96      FILL_HOMOG_STATS - Fill STATS buffs for homogs
```

[illegible]



```
0000 1 .TITLE HOMOG - MONITOR Homogeneous Class STATS Rtn
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 *
0000 8 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 * ALL RIGHTS RESERVED.
0000 11 *
0000 12 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 * TRANSFERRED.
0000 18 *
0000 19 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 * CORPORATION.
0000 22 *
0000 23 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *****
0000 27
0000 28 ++
0000 29 FACILITY: VAX/VMS MONITOR Utility
0000 30
0000 31 ABSTRACT:
0000 32
0000 33 This module fills all the STATS buffers for homogeneous
0000 34 STANDARD classes of the MONITOR utility.
0000 35
0000 36 ENVIRONMENT:
0000 37
0000 38 Unprivileged user mode, runs at AST level.
0000 39
0000 40 AUTHOR: Thomas L. Cafarella, April, 1983
0000 41
0000 42 MODIFIED BY:
0000 43
0000 44 V03-003 TLC1072 Thomas L. Cafarella 17-Apr-1984 11:00
0000 45 Add volume name to DISK display.
0000 46
0000 47 V03-002 TLC1063 Thomas L. Cafarella 3-Apr-1984 13:00
0000 48 Add check to ensure that a counter which is re-initiated to
0000 49 zero will not cause an **** to be displayed.
0000 50
0000 51 V03-001 TLC1061 Thomas L. Cafarella 18-Mar-1984 11:00
0000 52 Identify dual-path disks by allocation class.
0000 53
0000 54 V03-001 TLC1060 Thomas L. Cafarella 12-Mar-1984 11:00
0000 55 Make multi-file summary work for homogeneous classes.
0000 56
0000 57 --
```



- MONITOR Homogeneous Class STATS Rtn<sup>6 2</sup>

VAX/VMS Macro V04-00  
[MONITOR.SRC]HOMOG.MAR;1

Page 2  
(1)

```

0000 58
0000 59 :
0000 60 : MACROS:
0000 61 :
0000 62 :
0000 63 :
0000 64 : Local Macro Definitions
0000 65 :
0000 66 :
0000 67 :
0000 68 : ALLOC Macro - Dynamically allocate space on the stack.
0000 69 :
0000 70
0000 71 .MACRO ALLOC LENGTH,RSLDESC,RSLBUF
0000 72 .SUBL #<LENGTH+3>&<^C3>,SP
0000 73 .IF NB,RSLBUF
0000 74 .MOVL SP,RSLBUF
0000 75 .ENDC
0000 76 .PUSHL SP
0000 77 .PUSHL #LENGTH
0000 78 .MOVL SP,RSLDESC
0000 79 .ENDM ALLOC
0000 80

```

[illegible]

HOMOG  
V04-000

- MONITOR Homogeneous Class STATS Rtn<sup>H 2</sup>  
DECLARATIONS

16-SEP-1984 02:05:50 VAX/VMS Macro V04-00  
5-SEP-1984 02:00:46 [MONITOR.SRC]HOMOG.MAR;1

Page 3  
(2)

```
0000 82 .SBTTL DECLARATIONS
0000 83 .PSECT MONDATA,QUAD,NOEXE
0000 84 :
0000 85 : INCLUDE FILES:
0000 86 :
0000 87 :
0000 88 $CDBDEF ; Define Class Descriptor Block
0000 89 $CDXDEF ; Define CDB Extension
0000 90 $IDBDEF ; define item descriptor block offsets
0000 91 $MBPDEF ; Define Monitor Buffer Pointers
0000 92 $MONDEF ; Monitor Recording File Definitions
0000 93 $SCBDEF ; Define STATS Control Block
0000 94 $TM1DEF ; Define temporary storage offsets
```



```
0000 96 .SBTTL FILL_HOMOG_STATS - Fill STATS buffs for homogs
00000000 97 .PSECT $$MONCODE,ROWRT,EXE
0000 98
0000 99 :++
0000 100
0000 101 : FUNCTIONAL DESCRIPTION:
0000 102
0000 103 : FILL_HOMOG_STATS
0000 104
0000 105 : This routine fills all the STATS buffers for the
0000 106 : class indicated by CDBPTR. The SCB (STATS Control
0000 107 : Block) Table and the Element ID Table are also
0000 108 : updated. These tables maintain information about
0000 109 : the elements of this homogeneous class. An "element"
0000 110 : is, for example, a particular disk in the DISK class.
0000 111 : There is a STATS buffer for each item defined for the
0000 112 : class. An "item" is, for example, operation count
0000 113 : for the DISK class.
0000 114
0000 115 : INPUTS:
0000 116
0000 117 : 4(AP) - address of CURRENT collection buffer
0000 118
0000 119 : 8(AP) - address of PREVIOUS collection buffer
0000 120
0000 121 : IMPLICIT INPUTS:
0000 122
0000 123 : OUTPUTS:
0000 124
0000 125 : ALL STATS buffers for this homogeneous class filled.
0000 126
0000 127 : IMPLICIT OUTPUTS:
0000 128
0000 129 : CDB$E_COUNT and CDX$W_CUMELCT established for the current interval.
0000 130 : Element ID Table and SCB (STATS Control Block) updated.
0000 131
0000 132 : ROUTINE VALUE:
0000 133
0000 134 : RO = $$$_NORMAL
0000 135
0000 136 : SIDE EFFECTS:
0000 137
0000 138 : none
0000 139
0000 140 :--
0000 141
```

```

      OFFC 0000 143
      0000 144 .ENTRY FILL_HOMOG_STATS, ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
      0002 145
56 00000000'EF D0 0002 146      MOVL CDBPTR,R6 ; Load CDB addr
      57 32 A6 D0 0009 147      MOVL CDBSA_CDX(R6),R7 ; Load CDX addr
      58 04 AC D0 000D 148      MOVL 4(AP),R8 ; Load CURRENT coll buff addr
      0011 149      ALLOC TMP$K_SIZE,R0,R9 ; Allocate local temp storage
18 A6 0A A7 3C 001E 150      MOVZWL CDX$W_CUMELCT(R7), - ; Load element count for display
      0023 151      CDB$E_COUNT(R6)
      58 0D C0 0023 152      ADDL2 #MNR_CL$K_HSIZE,R8 ; Point to CURR coll buff prologue
      69 68 D0 0026 153      MOVL MNR_HOM$E_LCT(R8), - ; Load current buffer data block count
      0029 154      TMP$E_DBCT(R9)
      03 12 0029 155      BNEQ SS ; Br if have some
      008A 31 002B 156      BRW UPDATE_SCB_FLAGS ; Else skip past ID Table update
      002E 157 SS:
04 A9 58 08 C0 002E 158      ADDL2 #MNR_HOM$K_PSIZE,R8 ; Point to first data block
      0031 159      MOVZWL CDB$W_BLKLEN(R6), - ; Get data block length
      0036 160      TMP$E_DBLEN(R9)
08 A9 0A A7 3C 0036 161      MOVZWL CDX$W_CUMELCT(R7), - ; Load number of ID Table elements
      003B 162      TMP$E_ELIDCT(R9)
      5A 09 A7 9A 003B 163      MOVZBL CDX$E_ELIDLEN(R7),R10 ; Get element ID length
      003F 164
```



```
003F 166 :  
003F 167 : Loop through all data blocks in the CURRENT collection buffer.  
003F 168 : For each element (represented by a data block), try to find  
003F 169 : a match in the element ID table. The ID table represents elements  
003F 170 : which have been monitored for this request. On the first time  
003F 171 : through this routine, the table will be empty. The element ID table  
003F 172 : has several other associated tables, namely the STATS control  
003F 173 : block (SCB) table, and all the transformation buffers (STATS, MIN,  
003F 174 : MAX, SUM, PCSTATS, PCMIN, PCMAX and PCSUM). Each of these tables/  
003F 175 : buffers has one element for each monitored element (i.e., disk for  
003F 176 : the DISK class). The current number of elements in each of the tables  
003F 177 : is represented by CDX$W_CUMELCT.  
003F 178 :  
003F 179 :  
OC A9 D4 003F 180 CLRL TMP$$_DBIDX(R9) ; Clear data block index  
5B OC A7 D0 0042 181 10$: MOVL CDX$$_ELIDTABLE(R7),R11 ; Load Element ID Table addr  
10 A9 94 0042 182 CLRB TMP$$_FOUND(R9) ; Clear "element found" indicator  
54 D4 0049 184 CLRL R4 ; Clear element ID table index  
55 08 A9 D0 004B 185 MOVL TMP$$_ELIDCT(R9),R5 ; Load number of elements in ID table  
1E 13 004F 186 BEQL 40$ ; Br if table is empty  
57 5A D0 0051 187 MOVL R10,R7 ; Borrow R7 to hold elt id length  
03 4B A6 06 E1 0054 188 BBC #CDB$$_DISKAC, - ; Branch if no allocation class in name  
0059 189 CDB$$_FLAGS(R6),20$ ;  
00AF 30 0059 190 BSBW SHORTEN_DISKNAM ; Shorten element name for DISK  
005C 191 20$: CMPC3 R7,(R8),(R11) ; Match current element in table ?  
6B 68 57 29 005C 192 BNEQU 30$ ; Br if not  
10 A9 01 90 0060 193 MOVB #1,TMP$$_FOUND(R9) ; Yes -- indicate so  
07 11 0066 195 BRB 40$ ; ... and terminate loop  
5B 5A C0 0068 196 30$: ADDL2 R10,R11 ; Point to next element ID  
ED 54 55 F2 006B 198 AOBLS R5,R4,20$ ; Loop through element ID table  
006F 199  
006F 200 :  
006F 201 : At this point the entire element ID table has been scanned for a  
006F 202 : match to the current element in the CURRENT collection buffer.  
006F 203 :  
006F 204 :  
57 32 A6 D0 006F 205 40$: MOVL CDB$$_CDX(R6),R7 ; Re-load CDX addr  
06 10 A9 E8 0073 206 BLBS TMP$$_FOUND(R9),50$ ; Branch if element found in table  
54 0A A7 3C 0077 207 MOVZWL CDX$$_CUMELCT(R7),R4 ; Get next available element index  
007B 208  
007B 209  
007B 210 :  
007B 211 : NOTE -- if R4 is greater than or equal to MAXELTS, issue warning msg  
007B 212 : and simply branch to look at next coll buff data block.  
007B 213 :  
6B 10 007B 214  
007D 215 BSBW CHECK_TAB_SPACE ; Check if table space exhausted  
007D 216 ; NOTE -- if so, MONITOR request  
007D 217 ; ... is terminated  
007D 218  
55 54 03 C5 007D 219 50$: MULL3 #SCB$$_SIZE,R4,R5 ; Get SCB offset from index  
55 10 B745 9E 0081 220 MOVAB @CDX$$_SCBTABLE(R7)[R5],R5 ; Get SCB address  
0086 221  
0086 222
```

HOMOG  
V04-000

- MONITOR Homogeneous Class STATS Rtn L 2  
FILL\_HOMOG\_STATS - Fill STATS buffs for 16-SEP-1984 02:05:50 VAX/VMS Macro V04-00  
5-SEP-1984 02:00:46 [MONITOR.SRC]HOMOG.MAR;1

Page 7  
(6)

```
00 02 A5 00 E2 0086 223 BBSS #SCBSV_CURRENT, - ; Set 'current' bit indicating this
                                0088 224 SCBSB_FLAGS(R5),60$ ; element in ID table was in CURR buff
                                0088 225
                                0088 226 60$:
05 10 A9 E8 008B 227 BLBS TMP$B_FOUND(R9),70$ ; Branch if element found in table
```



```
008F 229 :  
008F 230 : Element in CURRENT buffer was NOT found in the element ID table.  
008F 231 : Add a new element to the table.  
008F 232 :  
008F 233 :  
0068 30 008F 234 BSBW ADD_NEW_ELT ; Add elt to table  
1B 11 0092 235 ; NOTE -- several registers altered  
0092 236 BRB 80$ ; Go look at next coll buff data block  
0094 237  
0094 238 :  
0094 239 : Element in CURRENT was found in the element ID table.  
0094 240 :  
0094 241 :  
0094 242 70$:  
0094 243  
65 5B 65 3C 0094 244 MOVZWL SCBSW_DBIDX(R5),R11 ; Get data block index for prev buff  
65 0C A9 B0 0097 245 MOVW TMP$L_DBIDX(R9),SCBSW_DBIDX(R5) ; Save curr index for next int  
OF 02 A5 01 E1 009B 246 BBC #SCBSV_ACTIVE, - ; Done with this elt if not active  
009B 247 SCBSB_FLAGS(R5),80$  
00A0 248  
00A0 249  
00A0 250 :  
00A0 251 : This element is active. Call routine to actually fill the STATS buffers,  
00A0 252 : given the element ID table index and the addresses of this element's  
00A0 253 : data blocks for both CURRENT and PREVIOUS collection buffers.  
00A0 254 :  
00A0 255 :  
5B 04 A9 C4 00A0 256 MULL2 TMP$L_DBLEN(R9),R11 ; Get data block offset from index  
52 08 AC D0 00A4 257 MOVL 8(AP),R2 ; Get ptr to PREVIOUS coll buff  
5B 15 A24B 9E 00A8 258 MOVAB <MNR_CLSSK_HSIZE+MNR_HOMSK_PSIZE>(R2)[R11],R11 ; Compute PREVIOUS data block addr  
00AD 259  
00AD 260  
6C 10 00AD 261 BSBW HOMOG_STATS ; Fill STATS buffs for all req'd items  
00AF 262 ; NOTE -- this subrtn destroys  
00AF 263 ; ... R0-R3 and R5  
00AF 264  
00AF 265 80$:  
00AF 266  
58 04 A9 C0 00AF 267 ADDL2 TMP$L_DBLEN(R9),R8 ; Point to next data block  
8A 0C A9 69 F2 00B3 268 AOBLS TMP$L_DBCT(R9), - ; Loop once for each elt in CURR buff  
00B3 269 TMP$L_DBIDX(R9),10$  
00B8 270  
00B8 271  
00B8 272
```

```
00B8 274 :  
00B8 275 : Now go through entire SCB (STATS control block) Table, setting  
00B8 276 : the SCBSV_ACTIVE bit for all elements which have SCBSV_CURRENT  
00B8 277 : set. Clear all the SCBSV_CURRENT bits as you go to prepare for  
00B8 278 : the next collection interval. The underlying assumption is that  
00B8 279 : all elements which were in the current collection are "active."  
00B8 280 : For elements with SCBSV_CURRENT not set, clear SCBSV_ACTIVE  
00B8 281 : and, in addition, zero the STATS buffers if SCBSV_ACTIVE had  
00B8 282 : been set (this is an element which has just gone inactive.)  
00B8 283 :  
00B8 284 :  
00B8 285 UPDATE_SCB_FLAGS:  
00B8 286 :  
59 10 A7 D0 00B8 287 MOVL CDX$A_SCBTABLE(R7),R9 ; Get SCB table addr  
5A 0A A7 3C 00BC 288 MOVZWL CDX$W_CUMELCT(R7),R10 ; ... and no. of elements in it  
1E 13 00C0 289 BEQL FHS_RET ; Quit if empty  
54 D4 00C2 290 CLRL R4 ; Init element number  
00C4 291 10$:  
0B 02 A9 00 E4 00C4 292 BBSC #SCBSV_CURRENT, - ; Br if current set  
00C9 293 SCBSB_FLAGS(R9),20$ ; ... Always clear current  
0B 02 A9 01 E5 00C9 294 BBCC #SCBSV_ACTIVE, - ; Br if active clear  
00CE 295 SCBSB_FLAGS(R9),30$ ; ... Always clear active  
58 D4 00CE 296 CLRL R8 ; Indicate "clear STATS buffers"  
49 10 00D0 297 BSBB HOMOG_STATS ; Clear all STATS buffs for this elt  
00D2 298 : NOTE -- this subrtn destroys  
00D2 299 : ... R0-R3 and R5  
05 11 00D2 300 BRB 30$ ; Go process next element  
00D4 301 20$:  
00 02 A9 01 E2 00D4 302 BBSS #SCBSV_ACTIVE, - ; Set active  
00D9 303 SCBSB_FLAGS(R9),30$  
00D9 304 30$:  
59 03 C0 00D9 305 ADDL2 #SCBSK_SIZE,R9 ; Point to next SCB  
E4 54 5A F2 00DC 306 AOBLSS R10,R4,10$ ; Loop back for next element  
00E0 307 :  
50 00000000'8F D0 00E0 308 FHS_RET:  
04 00E0 309 MOVL #SS$_NORMAL,R0 ; Normal status  
00E7 310 RET ; Return  
00E8 311
```



```
00E8 313
00E8 314 CHECK_TAB_SPACE:
00E8 315
54 00000000'8F D1 00E8 316 CMPL #MAXELTS,R4 ; Have we run out of table space?
08 14 00EF 317 BGTR 10$ ; Br if not
50 00000000'8F D0 00F1 318 MOVL #MNR$_TABLEFULL,R0 ; Yes -- return error
04 00F8 319 RET ; .....
05 00F9 320 10$: ;
00F9 321 RSB ; Return to caller
00FA 322
00FA 323
00FA 324 ADD_NEW_ELT: ; Add new element to ELIDTABLE
00FA 325
0A A7 B6 00FA 326 INCW CDX$_CUMELCT(R7) ; Count the new element
65 0C A9 B0 00FD 327
0101 328 MOVW TMP$_DBIDX(R9), - ; Set data block index ...
0101 329 SCB$_DBIDX(R5) ; ... for use next interval
OC B744 54 5A C4 0101 330 MULL2 R10,R4 ; Compute offset to new table entry
68 5A 28 0104 331 MOVC3 R10,(R8),@CDX$_ELIDTABLE(R7)[R4] ; Move new element ID into the table
05 010A 332 RSB ; Return
010B 333
010B 334
010B 335
010B 336 SHORTEN_DISKNAM: ; Possibly shorten DISK element name
010B 337
03 4B A6 07 E1 010B 338 BBC #CDB$_DISKVN, - ; Branch if no volume name in disk name
0110 339 CDB$_FLAGS(R6),10$
57 0C C2 0110 340 SUBL2 #12,R7 ; Shorten length for compare so
0113 341 ; ... volume name will not be compared
0113 342 10$:
0113 343 TSTB (R8) ; Allocation class zero ?
03 13 0115 344 BEQL 20$ ; Br if so
57 08 C2 0117 345 SUBL2 #8,R7 ; Shorten length for compare so
011A 346 ; ... node name will not be compared
011A 347 20$:
05 011A 348 RSB ; Return
```

```
011B 350 :  
011B 351 : HOMOG_STATS - Fill the STATS buffer for each requested data item  
011B 352 : (e.g., disk operation count) for the current element  
011B 353 : (e.g., DBA0).  
011B 354 :  
011B 355 : Register Inputs:  
011B 356 :  
011B 357 : R4 = element index of current element  
011B 358 : R6 = CDB address  
011B 359 : R7 = CDX address  
011B 360 : R8 = address of data block for CURRENT coll buffer,  
011B 361 : OR 0, if STATS buffers are to be cleared.  
011B 362 : R10 = element ID length (unused if R8 = 0)  
011B 363 : R11 = address of data block for PREVIOUS coll buffer  
011B 364 : (unused if R8 = 0)  
011B 365 :  
011B 366 : Implicit Inputs:  
011B 367 :  
011B 368 : Volatile registers: R0, R1, R2, R3, R5  
011B 369 :  
011B 370 : Implicit outputs:  
011B 371 :  
011B 372 : ALL STATS buffers updated.  
011B 373 :  
011B 374 :  
011B 375 : HOMOG_STATS:  
011B 376 :  
0600 8F B2 011B 377 : PUSH R9,R10 ; Save regs  
58 D5 011B 378 : TSTL R8 ; Data block ptr present?  
14 12 0121 379 : BNEQ 20$ ; Br if yes  
0123 380 :  
0123 381 : Special case: clear STATS buffers for this element  
0123 382 :  
0123 383 :  
50 06 A7 9A 0123 384 : MOVZBL CDX$B_IDISCT(R7),R0 ; Get number of STATS buffers  
51 2E A6 D0 0127 385 : MOVL CDB$A_BUFFERS(R6),R1 ; Get addr of first MBP ptr  
012B 386 :  
52 81 D0 012B 387 : MOVL (R1)+,R2 ; Get MBP pointer  
08 B244 D4 012E 388 : CLRL @MBP$A_STATS(R2)[R4] ; Clear STATS for this item & element  
F6 50 F5 0132 389 : SOBGTR R0,10$ ; Loop back for next STATS buffer  
24 11 0135 390 : BRB 40$ ; Go to common return
```



```

52 58 5A C1 0137 392 20$:
53 5B 5A C1 0137 393      ADDL3 R10,R8,R2      ; Point to 1st item in CURR data block
55 55 D4 013B 394      ADDL3 R10,R11,R3     ; Point to 1st item in PREV data block
      013F 395      CLRL R5                ; Init index to MBP pointers
      0141 396
      0141 397
      0141 398 ; Use FFS instruction to scan the item number bits in the CDX
      0141 399 ; When an item number is found, compute the STATS value for the item.
      0141 400
      0141 401
      51 10 D0 0141 402      MOVL #CDX$S_IBITS,R1 ; Init bit field size
      50 D4 0144 403      CLRL R0                ; Init start position
59 67 51 50 EA 0146 404 30$:
      0146 405      FFS R0,R1,CDX$W_IBITS(R7),R9 ; Search for next item number
      014B 406      ; R9 contains item number if found
      0E 13 014B 407      BEQL 40$              ; Branch if none found
      014D 408
      11 10 014D 409      BSBB STORE_STATS      ; Go store stats for item in R9
      014F 410      ; NOTE -- this subrtn destroys R10
      014F 411
      55 D6 014F 412      INCL R5                ; Index to next sequential MBP ptr
      0151 413
      50 59 01 0151 414      ADDL3 #1,R9,R0      ; Compute next starting ...
      51 10 50 C3 0155 415      SUBL3 R0,#CDX$S_IBITS,R1 ; ... position and field size
      EB 11 0159 416      BRB 30$              ; Go search rest of bit string
      015B 417 40$:
      015B 418
      0600 8F BA 015B 419      POPR #^M<R9,R10> ; Restore regs
      05 015F 420      RSB                      ; Return
      0160 421
      0160 422
      0160 423 STORE_STATS:
      0160 424
      5A 1C B649 9A 0160 425      MOVZBL @CDB$A_ITMSTR(R6)[R9],R10 ; Get IDB item number
      5A 11 C4 0165 426      MULL2 #IDB$K_ILENGTH,R10 ; Compute index into IDB table
5A 00000000'EF4A 9E 0168 427      MOVAB PERF$TABLE[R10],R10 ; Address of IDB for this item
0000'8F 0A AA B1 0170 428      CMPW IDB$W_TYPE(R10),#COUNT_TYPE ; Is this item a count?
      11 12 0176 429      BNEQ 10$              ; No -- assume level type
      5A 2E B645 D0 0178 430      MOVL @CDB$A_BUFFERS(R6)[R5],R10 ; Get MBP pointer
08 BA44 6249 6349 C3 017D 431      SUBL3 (R3)[R9],(R2)[R9],- ; Compute STATS for this count item
      0185 432      @MBP$A_STATS(R10)[R4]
      0D 18 0185 433      BGEQ 30$              ; Br if difference OK
      05 11 0187 434      BRB 20$              ; Ctr was re-init'ed; treat like level
      0189 435 10$:
      5A 2E B645 D0 0189 436      MOVL @CDB$A_BUFFERS(R6)[R5],R10 ; Get MBP pointer
      018E 437 20$:
08 BA44 6249 D0 018E 438      MOVL (R2)[R9],@MBP$A_STATS(R10)[R4] ; Store STATS for this level item
      0194 439 30$:
      05 0194 440      RSB                      ; Return
      0195 441
      0195 442 .END
```

HOMOG  
Symbol table

- MONITOR Homogeneous Class STATS Rtn

E 3

16-SEP-1984 02:05:50 VAX/VMS Macro V04-00  
5-SEP-1984 02:00:46 [MONITOR.SRC]HOMOG.MAR;1

Page 13  
(12)

ADD_NEW_ELT	= 000000FA	R	02	CDB\$V_PERCENT	= 00000000		
ALL_STAT	= 00000000			CDB\$V_QFILLER	= 00000002		
AVE_STAT	= 00000002			CDB\$V_STD	= 00000004		
CDB	= 00000000			CDB\$V_SWAPBUF	= 00000001		
CDB\$A_BUFFERS	= 0000002E			CDB\$V_SYSCLS	= 00000008		
CDB\$A_CDX	= 00000032			CDB\$V_UNIFORM	= 00000002		
CDB\$A_CHDHDR	= 0000004F			CDB\$V_WIDE	= 00000008		
CDB\$A_FAOCTR	= 00000004			CDB\$W_BLKLEN	= 00000020		
CDB\$A_ITMSTR	= 0000001C			CDB\$W_DISPCTL	= 00000036		
CDB\$A_POSTCOLL	= 00000026			CDB\$W_QFLAGS	= 00000045		
CDB\$A_PRECOLL	= 00000022			CDB\$W_QFLAGS_CUR	= 00000049		
CDB\$A_SUMBUF	= 0000000C			CDB\$W_QFLAGS_DEF	= 00000047		
CDB\$A_TITLE	= 00000010			CDBPTR	*****	X	02
CDB\$B_FAOPRELEN	= 00000041			CDB_EXT	= 00000000		
CDB\$B_FAOSEGLEN	= 00000040			CDX\$A_DISPFAO	= 0000002C		
CDB\$B_ST	= 00000042			CDX\$A_DISPNAME	= 00000028		
CDB\$B_ST_CUR	= 00000044			CDX\$A_ELIDTABLE	= 0000000C		
CDB\$B_ST_DEF	= 00000043			CDX\$A_ILOOKTAB	= 00000024		
CDB\$K_SIZE	= 00000053			CDX\$A_SCBTABLE	= 00000010		
CDB\$L_BUFFERS	= 0000002A			CDX\$A_SELIDTABLE	= 00000018		
CDB\$L_ECOUNT	= 00000018			CDX\$B_ELIDLEN	= 00000009		
CDB\$L_FAOCTR	= 00000000			CDX\$B_IDISCONSEC	= 00000007		
CDB\$L_FLAGS	= 00000048			CDX\$B_IDISCT	= 00000006		
CDB\$L_ICOUNT	= 00000014			CDX\$B_IDISINDEX	= 00000008		
CDB\$L_MIN	= 00000038			CDX\$K_SIZE	= 00000030		
CDB\$L_RANGE	= 0000003C			CDX\$K_DCOUNT	= 0000001C		
CDB\$L_SUMBUF	= 00000008			CDX\$K_PREV_DCT	= 00000020		
CDB\$M_CPU	= 00000002			CDX\$K_SELIDTABLE	= 00000014		
CDB\$M_CPU_COMB	= 00000008			CDX\$S_CDB_EXT	= 00000030		
CDB\$M_CTPRES	= 00000001			CDX\$S_IBITS	= 00000010		
CDB\$M_DISABLE	= 00000200			CDX\$W_CUMELCT	= 0000000A		
CDB\$M_DISKAC	= 00000040			CDX\$W_IBITS	= 00000000		
CDB\$M_DISKVN	= 00000080			CDX\$W_IBITS_CUR	= 00000004		
CDB\$M_EXPLIC	= 00001000			CDX\$W_IBITS_DEF	= 00000002		
CDB\$M_HOMOG	= 00000020			CHECK_TAB_SPACE	= 000000E8	R	02
CDB\$M_KUNITS	= 00000400			CLASS_HDR	= 00000000		
CDB\$M_PERCENT	= 00000001			COUNT_TYPE	*****	X	02
CDB\$M_STD	= 00000010			CUR_STAT	= 00000001		
CDB\$M_SWAPBUF	= 00000002			DEF\$A_DISP	= 0000000C		
CDB\$M_SYSCLS	= 00000100			DEF\$A_REC	= 00000004		
CDB\$M_UNIFORM	= 00000004			DEF\$A_SUMM	= 00000014		
CDB\$M_WIDE	= 00000800			DEF\$K_DISP	= 00000008		
CDB\$S_CDB	= 00000053			DEF\$K_REC	= 00000000		
CDB\$S_FILLER	= 00000013			DEF\$K_SUMM	= 00000010		
CDB\$S_FLAGS	= 00000004			DEF\$S_DEF_DESC	= 00000018		
CDB\$S_QFILLER	= 0000000E			DEF_DESC	= 00000000		
CDB\$S_QFLAGS	= 00000002			FHS_RET	= 000000E0	R	02
CDB\$V_CPU	= 00000001			FILE_HDR	= 00000000		
CDB\$V_CPU_COMB	= 00000003			FILL_HOMOG_STATS	= 00000000	RG	02
CDB\$V_CTPRES	= 00000000			HOMOG_STATS	= 0000011B	R	02
CDB\$V_DISABLE	= 00000009			HOM_CLASS_PRE	= 00000000		
CDB\$V_DISKAC	= 00000006			IDB	= 00000000		
CDB\$V_DISKVN	= 00000007			IDB\$A_ADDR	= 0000000C		
CDB\$V_EXPLIC	= 0000000C			IDB\$A_LNAME	= 00000004		
CDB\$V_FILLER	= 0000000D			IDB\$A_SNAME	= 00000000		
CDB\$V_HOMOG	= 00000005			IDB\$B_FLAGS	= 00000010		
CDB\$V_KUNITS	= 0000000A			IDB\$K_ILENGTH	= 00000011		



HOMOG  
Symbol table

F 3  
- MONITOR Homogeneous Class STATS Rtn

16-SEP-1984 02:05:50 VAX/VMS Macro V04-00  
5-SEP-1984 02:00:46 [MONITOR.SRC]HOMOG.MAR;1

Page 14  
(12)

IDBSM\_PCNT = 00000001  
IDBSS\_FILLER = 00000007  
IDBSS\_FLAGS = 00000001  
IDBSS\_IDB = 00000011  
IDBSV\_FILLER = 00000001  
IDBSV\_PCNT = 00000000  
IDBSW\_ISIZE = 00000008  
IDBSW\_TYPE = 0000000A  
MAXELTS \*\*\*\*\*  
MAX\_STAT = 00000004  
MBP = 00000000  
MBPSA\_ADDR = 00000018  
MBPSA\_B1ST = 00000004  
MBPSA\_BA = 00000000  
MBPSA\_BUFF1ST = 00000004  
MBPSA\_BUFFERA = 00000000  
MBPSA\_BUFFERB = 00000004  
MBPSA\_DATA = 00000008  
MBPSA\_DIFF = 0000000C  
MBPSA\_MAX = 00000010  
MBPSA\_MIN = 0000000C  
MBPSA\_ORDER = 00000010  
MBPSA\_PCMAX = 00000020  
MBPSA\_PCMIN = 0000001C  
MBPSA\_PCSTATS = 00000018  
MBPSA\_PCSUM = 00000024  
MBPSA\_PID = 00000014  
MBPSA\_PR\_FAOSTK = 00000008  
MBPSA\_STATS = 00000008  
MBPSA\_SUM = 00000014  
MBPSK\_SIZE = 00000028  
MBPSS\_MBP = 00000028  
MBPSS\_MBP2 = 0000001C  
MBPSS\_MBP3 = 0000000C  
MBP2 = 00000000  
MBP3 = 00000000  
MIN\_STAT = 00000003  
MNR\$ TABLEFULL \*\*\*\*\*  
MNR\_CLSSB\_TYPE = 00000000  
MNR\_CLSSK\_HSIZE = 0000000D  
MNR\_CLSSQ\_STAMP = 00000003  
MNR\_CLSSS\_CLASS\_HDR = 0000000D  
MNR\_CLSSS\_FILLER = 0000000F  
MNR\_CLSSS\_FLAGS = 00000002  
MNR\_CLSSS\_STAMP = 00000008  
MNR\_CLSSV\_CONT = 00000000  
MNR\_CLSSV\_FILLER = 00000001  
MNR\_CLSSW\_FLAGS = 00000001  
MNR\_CLSSW\_RESERVED = 0000000B  
MNR\_HDR\$B\_TYPE = 00000000  
MNR\_HDR\$K\_CLASSBITS = 00000073  
MNR\_HDR\$K\_MAXCOMLEN = 0000003C  
MNR\_HDR\$K\_REVLEVELS = 00000083  
MNR\_HDR\$K\_SIZE = 00000103  
MNR\_HDR\$L\_FLAGS = 00000001  
MNR\_HDR\$L\_INTERVAL = 00000015

X 02

X 02

MNR\_HDR\$L\_RECCT = 00000029  
MNR\_HDR\$O\_CLASSBITS = 00000073  
MNR\_HDR\$O\_REVOCLSBITS = 00000019  
MNR\_HDR\$Q\_BEGINNING = 00000005  
MNR\_HDR\$Q\_ENDING = 0000000D  
MNR\_HDR\$S\_BEGINNING = 00000008  
MNR\_HDR\$S\_CLASSBITS = 00000010  
MNR\_HDR\$S\_COMMENT = 0000003C  
MNR\_HDR\$S\_ENDING = 00000008  
MNR\_HDR\$S\_FILE\_HDR = 00000103  
MNR\_HDR\$S\_FILLER = 00000020  
MNR\_HDR\$S\_FLAGS = 00000004  
MNR\_HDR\$S\_LEVEL = 00000008  
MNR\_HDR\$S\_REVOCLSBITS = 00000010  
MNR\_HDR\$S\_REVLEVELS = 00000080  
MNR\_HDR\$S\_TYPE = 00000008  
MNR\_HDR\$T\_COMMENT = 00000035  
MNR\_HDR\$T\_LEVEL = 0000002D  
MNR\_HDR\$T\_REVLEVELS = 00000083  
MNR\_HDR\$V\_FILLER = 00000000  
MNR\_HDR\$W\_COMLEN = 00000071  
MNR\_HOM\$K\_PSIZE = 00000008  
MNR\_HOM\$L\_ELCTCT = 00000000  
MNR\_HOM\$L\_RESERVED = 00000004  
MNR\_HOM\$S\_HOM\_CLASS\_PRE = 00000008  
MNR\_PRO\$B\_PRI = 0000000A  
MNR\_PRO\$K\_DSIZE = 0000003B  
MNR\_PRO\$K\_FSIZE = 00000040  
MNR\_PRO\$K\_PSIZE = 00000008  
MNR\_PRO\$K\_REVODSIZE = 00000033  
MNR\_PRO\$K\_REV1DSIZE = 0000003B  
MNR\_PRO\$L\_BIOCNT = 0000002F  
MNR\_PRO\$L\_CPUTIM = 0000002B  
MNR\_PRO\$L\_DIOCNT = 00000023  
MNR\_PRO\$L\_EFWM = 00000037  
MNR\_PRO\$L\_EPID = 00000033  
MNR\_PRO\$L\_IPID = 00000000  
MNR\_PRO\$L\_PAGEFLTS = 00000027  
MNR\_PRO\$L\_PCTINT = 00000004  
MNR\_PRO\$L\_PCTREC = 00000000  
MNR\_PRO\$L\_STS = 0000001F  
MNR\_PRO\$L\_UIC = 00000004  
MNR\_PRO\$O\_LNAME = 0000000B  
MNR\_PRO\$S\_LNAME = 00000010  
MNR\_PRO\$S\_PROCESS\_CLASS = 0000003B  
MNR\_PRO\$S\_PRO\_CLASS\_PRE = 00000008  
MNR\_PRO\$W\_GPGCNT = 0000001B  
MNR\_PRO\$W\_PPGCNT = 0000001D  
MNR\_PRO\$W\_STATE = 00000008  
MNR\_SYISB\_MPCPUS = 0000000D  
MNR\_SYISB\_TYPE = 00000000  
MNR\_SYISK\_BALSETMEM = 0000001E  
MNR\_SYISK\_CPUTYPE = 00000026  
MNR\_SYISK\_MPWHILIM = 00000022  
MNR\_SYISK\_NODENAME = 0000000E  
MNR\_SYISK\_SIZE = 0000002A  
MNR\_SYISL\_BALSETMEM = 0000001E

HOMOG  
Symbol table

G 3  
- MONITOR Homogeneous Class STATS Rtn

16-SEP-1984 02:05:50 VAX/VMS Macro V04-00  
5-SEP-1984 02:00:46 [MONITOR.SRC]HOMOG.MAR;1

Page 15  
(12)

MNR\_SYISL\_CPUTYPE = 00000026  
MNR\_SYISL\_MPWHILIM = 00000022  
MNR\_SYISQ\_BOOTTIME = 00000003  
MNR\_SYISS\_BOOTTIME = 00000008  
MNR\_SYISS\_FILLER = 0000000E  
MNR\_SYISS\_FLAGS = 00000002  
MNR\_SYISS\_NODENAME = 00000010  
MNR\_SYISS\_SYS\_INFO = 0000002A  
MNR\_SYISS\_TYPE = 00000008  
MNR\_SYIST\_NODENAME = 0000000E  
MNR\_SYISV\_CLUSMEM = 00000000  
MNR\_SYISV\_FILLER = 00000002  
MNR\_SYISV\_RESERVED1 = 00000001  
MNR\_SYISW\_FLAGS = 00000001  
MNR\_SYISW\_MAXPRCCT = 0000000B  
PERFTABLE = \*\*\*\*\*  
PROCDISPS = 00000005  
PROCESS\_CLASS = 00000000  
PRO\_CLASS\_PRE = 00000000  
QUALSA\_ALC = 00000064  
QUALSA\_AVE = 00000074  
QUALSA\_BEG = 00000004  
QUALSA\_BY\_NODE = 00000054  
QUALSA\_CLASS = 0000005C  
QUALSA\_COMM = 0000004C  
QUALSA\_CPU = 000000AC  
QUALSA\_CUR = 0000006C  
QUALSA\_DISP = 00000034  
QUALSA\_END = 0000000C  
QUALSA\_FLUSH = 0000001C  
QUALSA\_INP = 0000002C  
QUALSA\_INT = 00000014  
QUALSA\_ITEM = 000000BC  
QUALSA\_MAX = 00000084  
QUALSA\_MIN = 0000007C  
QUALSA\_PCEN = 000000B4  
QUALSA\_REC = 0000003C  
QUALSA\_SUMM = 00000044  
QUALSA\_TOPB = 0000009C  
QUALSA\_TOPC = 0000008C  
QUALSA\_TOPD = 00000094  
QUALSA\_TOPF = 000000A4  
QUALSA\_VIEW = 00000024  
QUALSL\_ALL = 00000060  
QUALSL\_AVE = 00000070  
QUALSL\_BEG = 00000000  
QUALSL\_BY\_NODE = 00000050  
QUALSL\_CLASS = 00000058  
QUALSL\_COMM = 00000048  
QUALSL\_CPU = 000000A8  
QUALSL\_CUR = 00000068  
QUALSL\_DISP = 00000030  
QUALSL\_END = 00000008  
QUALSL\_FLUSH = 00000018  
QUALSL\_INP = 00000028  
QUALSL\_INT = 00000010  
QUALSL\_ITEM = 000000B8

X 02

QUALSL\_MAX = 00000080  
QUALSL\_MIN = 00000078  
QUALSL\_PCEN = 000000B0  
QUALSL\_REC = 00000038  
QUALSL\_SUMM = 00000040  
QUALSL\_TOPB = 00000098  
QUALSL\_TOPC = 00000088  
QUALSL\_TOPD = 00000090  
QUALSL\_TOPF = 000000A0  
QUALSL\_VIEW = 00000020  
QUALSS\_QUALIFIER\_DESC = 000000C0  
QUALIFIER\_DESC = 00000000  
REG\_PROC = 00000000  
SCBSB\_FLAGS = 00000002  
SCBSK\_SIZE = 00000003  
SCBSS\_FILLER = 00000006  
SCBSS\_FLAGS = 00000001  
SCBSS\_STATS\_BLOCK = 00000003  
SCBSV\_ACTIVE = 00000001  
SCBSV\_CURRENT = 00000000  
SCBSV\_FILLER = 00000002  
SCBSW\_DBIDX = 00000000  
SHORTEN\_DISKNAME = 0000010B R 02  
SSS\_NORMAL = \*\*\*\*\* X 02  
STATS = 00000005  
STATS\_BLOCK = 00000000  
STORE\_STATS = 00000160 R 02  
SYS\_INFO = 00000000  
TEMP\_1\_BLOCK = 00000000  
TMP\$B\_FOUND = 00000010  
TMP\$K\_SIZE = 00000011  
TMP\$L\_DBCT = 00000000  
TMP\$L\_DBIDX = 0000000C  
TMP\$L\_DBLEN = 00000004  
TMP\$L\_ELIDCT = 00000008  
TMP\$S\_TEMP\_1\_BLOCK = 00000011  
TOPB\_PROC = 00000003  
TOPC\_PROC = 00000001  
TOPD\_PROC = 00000002  
TOPF\_PROC = 00000004  
UPDATE\_SCB\_FLAGS = 000000B8 R 02



+-----+  
! Psect synopsis !  
+-----+

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
MONDATA	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC QUAD
\$\$MONCODE	00000195 ( 405.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:01.43
Command processing	111	00:00:00.75	00:00:06.13
Pass 1	161	00:00:02.44	00:00:09.57
Symbol table sort	0	00:00:00.48	00:00:00.66
Pass 2	88	00:00:01.08	00:00:03.43
Symbol table output	39	00:00:00.34	00:00:01.12
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	432	00:00:05.22	00:00:22.36

The working set limit was 1200 pages.  
15442 bytes (31 pages) of virtual memory were used to buffer the intermediate code.  
There were 20 pages of symbol table space allocated to hold 326 non-local and 22 local symbols.  
442 source lines were read in Pass 1, producing 17 object records in Pass 2.  
16 pages of virtual memory were used to define 8 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
_\$255\$DUA28:[MONITOR.OBJ]MONLIB.MLB;1	7
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0
TOTALS (all libraries)	7

327 GETS were required to define 7 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:HOMOG/OBJ=OBJ\$:HOMOG MSRC\$:HOMOG/UPDATE=(ENH\$:HOMOG)+EXECML\$/LIB+LIB\$:MONLIB/LIB



0240 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

